1974

# An iterative method for the solution of the algebraic eigenvalue problem for Hermitian matrices

Robert Stephen Conker
*Iowa State University*

# An iterative method for the solution of the algebraic eigenvalue problem for Hermitian matrices

by

Robert Stephen Conker

A Dissertation Submitted to the
Graduate Faculty in Partial Fulfillment of
The Requirements for the Degree of
DOCTOR OF PHILOSOPHY

Major: Computer Science

Approved:

Signature was redacted for privacy.

In Charge of Major Work

Signature was redacted for privacy.

For the Major Department

Signature was redacted for privacy.

For the Graduate College

Iowa State University
Ames, Iowa
1974

TABLE OF CONTENTS

# I. INTRODUCTION

For the last decade a large amount of work has gone into the numerical solution of what is called the algebraic eigenvalue problem; that is, the calculation of those numbers $\lambda$ and vectors x such that

$$Ax = \lambda x \qquad (1.1)$$

for a given square matrix A.

Many methods exist which solve (1.1) depending on whether A is real or complex, Hermitian or non-Hermitian, and whether or not the matrix is in Hessenberg or full form. There have also been various methods developed depending on whether or not several or all of the eigenvalues and their respective eigenvectors are desired. A few methods, such as the Jacobi and Power methods, have fallen or are falling into disuse due to a slow rate of convergence, or inaccurate calculations. (4)

In addition, current approaches many times segment the problem into several distinct subproblems. For instance, the computation of all the eigenvalues and vectors of a symmetric matrix can be considered to be four distinct steps: (1.) Householder transformation of the matrix to tridiagonal form, (2.) diagonalizing the tridiagonal form, (3.) determination of the eigenvectors of the tridiagonal form, and (4.) back transforming these vectors to those of the original matrix.

After the determination of the eigenvalues, only a few eigenvectors may be desired. In this case, a method denoted inverse

iteration (or Rayleigh quotient inverse iteration) can be used.

A variation of Rayleigh quotient inverse iteration (frequently referred to as RQI) will be the focus of this thesis. In RQI, the iterative scheme computes successive vector iterates $\{x_m\}$ in the following manner: (8)

$$(A - \mu_m I)x_{m+1} = kx_m \tag{1.2}$$

where $\{x_m\}, m = 0,1,2,\ldots$ are considered to be normalized to length one, k a positive normalizing constant, and $\mu_m$ the Rayleigh quotient at the $m^{th}$ step, such that

$$\mu_m = \frac{x_m^* A x_m}{x_m^* x_m} = x_m^* A x_m . \tag{1.3}$$

In the case where an eigenvalue $\lambda$ has already been accurately determined by some other method, then $\mu_m = \lambda$ and is held constant.

Much of the work done in this area has been by J. H. Wilkinson, whose papers on the subject are extensive. P. Chuang (3) has developed an iterative method referred to loosely as a norm reduction method since its vector iterates are computed on the basis of the direct reduction of the norm of the residual vector at each step, where $r_m$ denotes the residual vector at the $m^{th}$ step and is defined to be

$$r_m = (A - \mu_m I)x_m . \tag{1.4}$$

The computation involves the successive solution of a system of linear equations defined by

$$(Q_m^2 - x_m x_m^* Q_m^2 - x_m x_m^*) x_{m+1} = k x_m \qquad (1.5)$$

where $Q_m = A - \mu_m I$. This system, requiring more computation than RQI, was derived with the viewpoint of obtaining high accuracy since the matrix A would not be transformed as it is in many other methods. The derivation rests on the consideration that each succeeding vector iterate may be represented as the normalized sum of the preceding vector and the difference (or incremental) vector. This incremental vector at the $m^{th}$ step was chosen orthogonal to the eigenvector iterate at the $m^{th}$ step. This was the basis of the derivation of (1.5) and will also be the basis of the derivation of the method presented in this thesis.

## II. DERIVATION OF THE METHOD

Let $g_m$ denote the incremental vector at the $m^{th}$ step. The iterative method of (1.2) can then be considered to involve the successive solution of the linear system

$$(A - \mu_m I)(x_m + g_m) = x_m \qquad (2.1)$$

where $x_m + g_m$ is unnormalized and

$$x_{m+1} = \frac{x_m + g_m}{||x_m + g_m||}$$

is the normalized vector iterate at the $m+1^{st}$ step.

Let $r_m$ be defined as $r_m = (A - \mu_m I)x_m$ and $r_{m+\frac{1}{2}}$ be defined as $r_{m+\frac{1}{2}} = (A - \mu_m I)x_{m+1}$. The matrix A will be considered Hermitian. For a Hermitian matrix, it was first shown in Chuang (3) that

$$||r_{m+1}||^2 = ||r_{m+\frac{1}{2}}||^2 - (\mu_{m+1} - \mu_m)^2. \qquad (2.2)$$

As stated before, each vector iterate in the sequence $\{x_m\}$, $m = 0,1,2,\ldots$ will be considered normalized to length one. And we shall require in the derivation that $g_m$ be orthogonal to $x_m$, or $x_m^* g_m = 0$.

Consider again (2.2). For the iterative method proposed by Chuang (3), $||r_{m+1}||^2$ was made to decrease by directly minimizing $||r_{m+\frac{1}{2}}||^2$. On the other hand, it seems plausible that $||r_{m+1}||^2$ might be reduced by finding some critical point of $(\mu_{m+1} - \mu_m)^2$. We shall in fact consider only the numerator of the linear expression

$(\mu_{m+1} - \mu_m)$ over a common denominator. Then

$$\mu_{m+1} - \mu_m = x^*_{m+1}Ax_{m+1} - x^*_m Ax_m = \frac{(x^*_m + g^*_m)A(x_m + g_m)}{||x_m + g_m||^2} - x^*_m Ax_m$$

$$= \frac{x^*_m Ax_m + x^*_m Ag_m + g^*_m Ax_m + g^*_m Ag_m - (1 + ||g_m||^2)x^*_m Ax_m}{1 + ||g_m||^2}$$

$$= \frac{x^*_m Ag_m + g^*_m Ax_m + g^*_m Ag_m - \mu_m ||g_m||^2}{1 + ||g_m||^2} \qquad (2.3)$$

Let $\{p_1, p_2, \ldots p_{n-1}\}$ be an arbitrary complex orthonormal basis for the vector space orthogonal to the $m^{th}$ iterate $x_m$. Since the change vector will be constructed orthogonal to $x_m$, then $g_m$ can be expressed as a linear combination of the vectors in the above basis. Therefore $g_m$ may be expressed as

$$g_m = Pc \qquad (2.4)$$

where $P = (p_1, p_2, \ldots p_{n-1})$ is an $n \times (n - 1)$ matrix and $c$ is the complex coefficient vector such that

$$c = \alpha + i\beta = \begin{pmatrix} \alpha_1 + i\beta_1 \\ \alpha_2 + i\beta_2 \\ \cdot \\ \cdot \\ \cdot \\ \alpha_{n-1} + i\beta_{n-1} \end{pmatrix}$$

Consequently the equation (2.3) can be rewritten as

$$\mu_{m+1} - \mu_m = \frac{f(c)}{1 + ||g_m||^2}$$

where

$$f(c) = x_m^* APc + (Pc)^* Ax_m + (Pc)^* APc - (Pc)^* Pc\mu_m. \qquad (2.5)$$

To find a critical point of f(c), it is necessary to solve the two linear systems

$$\frac{\partial f}{\partial \alpha_j} = 0 \qquad j = 1, 2, \ldots, n-1. \qquad (2.6)$$

$$\frac{\partial f}{\partial \beta_j} = 0$$

From (2.5) and the fact that $Pc = \sum_{k=1}^{n-1}(\alpha_k + i\beta_k)p_k$,

$$f(c) = x_m^* A\{\sum_{k=1}^{n-1}(\alpha_k + i\beta_k)p_k\} + \{\sum_{k=1}^{n-1}(\alpha_k + i\beta_k)p_k\}^* Ax_m$$

$$+ \{\sum_{k=1}^{n-1}(\alpha_k + i\beta_k)p_k\}^* A\{\sum_{k=1}^{n-1}(\alpha_k + i\beta_k)p_k\}$$

$$- \{\sum_{k=1}^{n-1}(\alpha_k + i\beta_k)p_k\}^* \{\sum_{k=1}^{n-1}(\alpha_k + i\beta_k)p_k\}\mu_m.$$

Then, $\frac{\partial f}{\partial \alpha_j} = x_m^* Ap_j + p_j^* Ax_m + \{\sum_{k=1}^{n-1}(\alpha_k + i\beta_k)p_k\}^* Ap_j$

$$+ p_j^* A\{\sum_{k=1}^{n-1}(\alpha_k + i\beta_k)p_k\} - p_j^* \{\sum_{k=1}^{n-1}(\alpha_k + i\beta_k)p_k\}\mu_m$$

$$- \{\sum_{k=1}^{n-1}(\alpha_k + i\beta_k)p_k\}^* p_j\mu_m.$$

$$= x_m^* A p_j + p_j^* A x_m + g_m^* A p_j + p_j^* A g_m - \mu_m\{(\alpha_j + i\beta_j) + (\alpha_j - i\beta_j)\}$$

$$= x_m^* A p_j + p_j^* A x_m + g_m^* A p_j + p_j^* A g_m - 2\mu_m\alpha_j = 0. \qquad (2.7)$$

And, $$\frac{\partial f}{\partial \beta_j} = ix_m^* A p_j - ip_j^* A x_m + \{\sum_{k=1}^{n-1}(\alpha_k + i\beta_k)p_k\}^* A i p_j$$

$$- ip_j^* A\{\sum_{k=1}^{n-1}(\alpha_k + i\beta_k)p_k\} - \{\sum_{k=1}^{n-1}(\alpha_k + i\beta_k)p_k\}^* i p_j \mu_m$$

$$+ ip_j^*\{\sum_{k=1}^{n-1}(\alpha_k + i\beta_k)p_k\}\mu_m$$

$$= ix_m^* A p_j - ip_j^* A x_m + ig_m^* A p_j - ip_j^* A g_m$$

$$- \mu_m\{(\alpha_j - i\beta_j)i - i(\alpha_j + i\beta_j)\}$$

$$= ix_m^* A p_j - ip_j^* A x_m + ig_m^* A p_j - ip_j^* A g_m - 2\mu_m\beta_j = 0. \quad (2.8)$$

Equation (2.8) may be multiplied by -i and added to equation (2.7) to produce one equation to solve for each $j = 1, 2, \ldots, n-1$, namely

$$\frac{\partial f}{\partial \alpha_j} - i\frac{\partial f}{\partial \beta_j} = 0. \qquad (2.9)$$

The above equation implies that the conditions of (2.6) are met and

(2.9) can then be reduced to

$$x_m^* A p_j + g_m^* A p_j - \mu_m c_j^* = 0$$

which for all j = 1, 2, . . . , n-1 becomes $(x_m^* + g_m^*)AP = \mu_m c^*$ or $P^* A(x_m + g_m) = \mu_m c$. With the addition of $x_m^*(x_m + g_m) = 1$, we have

$$\begin{pmatrix} P^* A \\ x_m^* \end{pmatrix}(x_m + g_m) = \begin{pmatrix} \mu_m c \\ 1 \end{pmatrix} .$$

Since $PP^* = I - x_m x_m^*$, then premultiplying by $(P, x_m)$ yields the following

$$(A - x_m x_m^* A + x_m x_m^*)(x_m + g_m) = \mu_m g_m + x_m$$

which becomes

$$(A - x_m x_m^* A - \mu_m I)(x_m + g_m) = -\mu_m x_m . \qquad (2.10)$$

Then given $x_m$, it is necessary to solve the linear system (2.10) to obtain $(x_m + g_m)$ which when normalized will produce $x_{m+1}$. This system can be solved using Gaussian elimination. But in Chapter VI, a method is devised whereby (2.10) can be solved in order n multiplications, a significant savings.

The method described above can be implemented by the following steps:

(1.) Choose an arbitrary initial starting vector $x_0$, such that $||x_0|| = 1$. Let $x_m = x_0$ where m = 0.

(2.) Calculate an acceptance threshold for the norm of the

residual vector ($t_1 = n \cdot ||A|| \cdot 10^{-p}$ where $10^{-p}$ represents the precision of the machine. See (3,10).)

(3.) Compute the Rayleigh quotient $\mu_m = x_m^* A x_m$.

(4.) Calculate $||r_m|| = ||Qx_m||$.

(5.) If $||r_m|| \leq t_1$ accept $x_m$ as an eigenvector and $\mu_m$ as an eigenvalue of A and go to step (9.); otherwise,

(6.) Form the matrix $H_m = A - x_m x_m^* A - \mu_m I$.

(7.) Solve the system $H_m(x_m + g_m) = -\mu_m x_m$ for $x_m + g_m$.

(8.) Normalize $x_m + g_m$: calculate $x_{m+1} = (x_m + g_m)/||x_m + g_m||$ and go to step (3.).

(9.) Reset m to 0 and choose a new starting vector such that $x_0$ is in the space orthogonal to all previous eigenvectors, $k \geq 1$. This $x_0$ will be chosen from one of the columns in the matrix $I - \sum_{j=1}^{k} v_j v_j^*$ such that $v_j$ is a normalized previously determined eigenvector. Go to step (3.) and continue this procedure until all eigenvectors have been determined.

RQI and the method given in (2.10) can be used as independent methods to determine the eigenvectors and their corresponding eigenvalues with no knowledge of any approximation. Also given an eigenvector, eigenvalue pair of moderate accuracy, these procedures may be used to correct the pair to somewhat the limits of machine precision. However, a third mode of operation, widely used, involves the determination of a very accurate eigenvalue, $\lambda$, by some reputable method (such as QR)

and then performing an iteration to determine an accurate eigenvector (7,9). In such a case, (2.1) and (2.10) are changed respectively to:

$$(A - \lambda I)(x_m + g_m) = x_m \qquad (2.11.a)$$

and

$$(A - x_m x_m^* A - \lambda I)(x_m + g_m) = -\mu_m x_m \qquad (2.11.b)$$

where $\lambda$ is a "good" eigenvalue and is held constant.

Notice that the Rayleigh quotient $\mu_m$ on the right hand side of (2.11.b) is not held constant. This is due to the change in the definitions of the residual norms. Now $r_m = (A - \lambda I)x_{m+1}$. Consequently $r_{m+1} = r_{m+\frac{1}{2}}$. With these changes in the residual norms, the proof of convergence of $\{||r_m||\}$ in (2.11.b) will be essentially the same as that given in the next chapter for method (2.10).

## III. CONVERGENCE

Since the norm of the residual vector is a continuous function of x, it is sufficient for convergence to show that every iteration reduces the norm. However, convergence to zero of the norm of the residual vector is what we are interested in proving. It can be seen from (2.2) that $||r_{m+1}||^2$ is always less than or equal to $||r_{m+\frac{1}{2}}||^2$. Therefore to prove convergence to zero it is necessary and sufficient to show that the following conditions are met:

$$\text{(a.)} \quad ||r_{m+\frac{1}{2}}|| \leq ||r_m|| \qquad m = 0, 1, 2, \ldots$$

$$\text{(b.)} \quad ||r_m|| \to 0 \quad \text{as } m \to \infty.$$

The following theorem will establish this.

Theorem 3.1: Let A be any n x n Hermitian matrix. Let $x_m \subset \{y: ||y||=1\}$ and let $\{H_m^{-1}\}$ exist for m any non-negative integer where $H_m = A - x_m x_m^* A - \mu_m I$ and $\mu_m = x_m^* A x_m$. Then the successive solution of (2.10) will cause conditions (a.) and (b.) to be satisfied.

Proof: Condition (a.)

Equation (2.10) can be expanded to obtain

$$Ax_m - x_m x_m^* Ax_m - \mu_m x_m + Ag_m - x_m x_m^* Ag_m - \mu_m g_m = -\mu_m x_m.$$

Regrouping and using the notation and conditions defined previously, we get

$$Q_m x_m + Q_m g_m = (x_m^* Ag_m) x_m.$$

Then $||r_{m+\frac{1}{2}}||^2 = ||Q_m x_{m+1}||^2 = \dfrac{||Q_m(x_m + g_m)||^2}{||x_m + g_m||^2} = \dfrac{||Q_m x_m + Q_m g_m||^2}{1 + ||g_m||^2}$

$$= \frac{||(x_m^* A g_m)x_m||^2}{1 + ||g_m||^2} \qquad (3.1)$$

Consider that $Ax_m = \mu_m x_m + r_m$ for any iteration m. Then $g_m^* Ax_m = \mu_m g_m^* x_m + g_m^* r_m = g_m^* r_m$ since $g_m$ and $x_m$ are orthogonal. This result can then be substituted into (3.1) such that

$$||r_{m+\frac{1}{2}}||^2 = \frac{|(x_m^* A g_m)|^2}{1 + ||g_m||^2} = \frac{|(g_m^* A x_m)^*|^2}{1 + ||g_m||^2} = \frac{|g_m^* A x_m|^2}{1 + ||g_m||^2}$$

$$= \frac{|g_m^* r_m|^2}{1 + ||g_m||^2} \leq \frac{||g_m||^2 ||r_m||^2}{1 + ||g_m||^2} \leq ||r_m||^2 \qquad (3.2)$$

Thus condition (a.) is satisfied.

Condition (b.): Condition (a.) shows that $\{||r_m||\}$ is monotonically decreasing. The norm of the residual vector therefore either approaches zero (satisfying condition (b.)) or it does not. Suppose the latter situation occurs. Then $||r_m|| \to \rho > 0$ as $m \to \infty$. This possibility can be separated into the following two cases.

(i) $\lim_{m\to\infty} ||r_m|| = \rho > 0$ and $g_m \to 0$.

In this case, $g_m \to 0$ implies $(\mu_{m+1} - \mu_m) \to 0$, so $\lim_{m\to\infty} ||r_{m+1}||^2 =$

$\lim_{m\to\infty}(||r_{m+\frac{1}{2}}||^2 - (\mu_{m+1} - \mu_m)^2) = \lim_{m\to\infty}||r_{m+\frac{1}{2}}||^2$. However,

$$\lim_{m\to\infty}||r_{m+\frac{1}{2}}||^2 \le \lim_{m\to\infty}\frac{||g_m||^2||r_m||^2}{1 + ||g_m||^2} = 0, \text{ so } \lim_{m\to\infty}||r_m||^2 = 0,$$

thus contradicting $\rho > 0$.

(ii) $\lim_{m\to\infty}||r_m|| = \rho > 0$ and $g_m \neq 0$.

Then there exists a convergent subsequence of vectors $\{x_k\} \subset \{x_m\}$, and a vector y (not an eigenvector) such that

$$\lim_{k\to\infty}\{x_k\} = y \text{ and } \lim_{k\to\infty}||r_k|| = \rho.$$

From the definition of $r_{m+\frac{1}{2}}$ given before,

$$||r_{m+\frac{1}{2}}||^2 = \frac{||r_m||^2 + 2\mathrm{Re}\{g_m^* Q_m r_m\} + ||Q_m g_m||^2}{1 + ||g_m||^2}.$$

From (3.2),

$$||r_{m+\frac{1}{2}}||^2 \le \frac{||g_m||^2||r_m||^2}{1 + ||g_m||^2}.$$

Together these relations imply that

$$2\mathrm{Re}\{g_m^* Q_m r_m\} + ||Q_m g_m||^2 < ||g_m||^2||r_m||^2. \qquad (3.3)$$

Since the incremental vector $g_k$ is a function of $x_k$,

$$\rho^2 = \lim_{k\to\infty}||r_{k+1}||^2 \le \lim_{k\to\infty}||r_{k+\frac{1}{2}}||^2$$

$$= \lim_{k\to\infty} \frac{||r_k||^2 + 2\mathrm{Re}\{g(x_k)^* Q_k r_k\} + ||Q_k g(x_k)||^2}{1 + ||g(x_k)||^2}$$

$$< \lim_{k\to\infty} \frac{||r_k||^2 + ||g(x_k)||^2 ||r_k||^2}{1 + ||g(x_k)||^2}$$

$$= \lim_{k\to\infty} ||r_k||^2 \left( \frac{1 + ||g(x_k)||^2}{1 + ||g(x_k)||^2} \right) = \rho^2$$

which implies $\rho^2 < \rho^2$, a contradiction. Therefore $\rho = 0$ and in all cases $\{||r_m||\}$ converges to zero. QED

It will be shown in the next chapter that this method is equivalent to Rayleigh quotient inverse iteration. Since Ostrowski (5) has shown RQI to converge cubically, the method derived here also exhibits cubic convergence where $H_m^{-1}$ exists.

## IV. EQUIVALENCE OF THE METHOD TO RQI

The purpose of this section is to prove the equivalence between RQI and the method derived in Chapter II. The importance is several-fold: (1.) The two methods can be interchanged depending on the use; (2.) the two methods will have contrasting incremental vectors, $g_m$, and yet will yield essentially the same solution (when normalized); (3.) the derived method will be established as a variation of inverse iteration (in particular cubic convergence will apply to it ).

Rayleigh quotient inverse iteration implies that the $m+1^{st}$ vector iterate before normalization is

$$x_m + g_m = (A - \mu_m I)^{-1} x_m \qquad (4.1)$$

assuming $A - \mu_m I$ is nonsingular.

The same $m+1^{st}$ unnormalized vector iterate will now be found for the linear system given in (2.10). Equation (2.10) implies that

$$x_{\bar{m}} + g_{\bar{m}} = -\mu_{\bar{m}}(A - x_{\bar{m}} x_{\bar{m}}^* A - \mu_{\bar{m}} I)^{-1} x_{\bar{m}} \qquad (4.2)$$

assuming $(A - x_m x_m^* A - \mu_m I)$ is nonsingular.

We will assume that $(A - \mu_m I)^{-1}$ exists and $\mu_m \neq 0$. Since $x_m x_m^* A$ is a simple product matrix, $(A - x_m x_m^* A - \mu_m I)^{-1}$ can be calculated by the method of completion (1).

Hence,

$$(A - x_m x_m^* A - \mu_m I)^{-1} = (A-\mu_m I)^{-1} + \frac{(A-\mu_m I)^{-1} x_m x_m^* A (A-\mu_m I)^{-1}}{1 - x_m^* A (A-\mu_m I)^{-1} x_m} \qquad (4.3)$$

Considering the denominator of the above and the identity

$$1 = x_m^*(A - \mu_m I)(A - \mu_m I)^{-1}x_m$$

$$= x_m^* A(A - \mu_m I)^{-1}x_m - \mu_m x_m^*(A - \mu_m I)^{-1}x_m$$

it is seen that

$$1 - x_m^* A(A - \mu_m I)^{-1}x_m = -\mu_m x_m^*(A - \mu_m I)^{-1}x_m \qquad (4.4)$$

Substituting (4.3) into (4.2) gives

$$x_m + g_m = -\mu_m\left(I + \frac{(A - \mu_m I)^{-1}x_m x_m^* A}{1 - x_m^* A(A - \mu_m I)^{-1}x_m}\right)(A - \mu_m I)^{-1}x_m,$$

and substituting (4.4) for the denominator of this expression yields

$$x_m + g_m = -\mu_m\left(I - \frac{(A - \mu_m I)^{-1}x_m x_m^* A}{\mu_m x_m^*(A - \mu_m I)^{-1}x_m}\right)(A - \mu_m I)^{-1}x_m$$

$$= -\mu_m(A - \mu_m I)^{-1}x_m + \frac{(A - \mu_m I)^{-1}x_m}{x_m^*(A - \mu_m I)^{-1}x_m}\{1 + \mu_m x_m^*(A - \mu_m I)^{-1}x_m\}$$

$$= \left(\frac{1}{x_m^*(A - \mu_m I)^{-1}x_m}\right)(A - \mu_m I)^{-1}x_m \qquad (4.5)$$

Thus (4.5) is a scalar multiple of the vector in (4.1) and when both are normalized the $m+1^{st}$ vector iterate $x_{m+1}$ will be plus or

minus the corresponding vector obtained from RQI. The same holds true for the residual vector $r_{m+1}$.

In particular, the norms of the residual vectors will be equal. Thus given a common starting vector, RQI and the method given in (2.10) will converge to the same eigenvector and at the same rate. We know from (5,6,9) that RQI converges cubically and from Chapter III that $||r_m||$ converges to zero if $H_m^{-1}$ exists. Thus equivalence of the two methods shows that both possess global convergence of residual norms to zero under the assumption that $H_m^{-1}$ exists, $\mu_m \neq 0$, and $(A - \mu_m I)^{-1}$ exists for all m = 0, 1, 2,.... Also convergence of the eigenvalues is cubic, where cubic convergence of eigenvalues can be construed to mean

$$\frac{(\mu_{m+1} - \lambda)}{(\mu_m - \lambda)^3} \rightarrow \gamma \quad (\gamma, \text{ a constant})$$

Another interesting equivalence of a different type exists between the system of (2.10) defined for the Algebraic Eigenvalue Problem, $Ax = \lambda x$, and the corresponding system which holds for the Generalized Eigenvalue Problem, $Ax = \lambda Bx$ where A and B are Hermitian matrices and B is positive definite. For the Generalized Eigenvalue Problem the system corresponding to (2.10) is

$$(A - Bx_m x_m^* A - \mu_m B)(x_m + g_m) = -\mu_m Bx_m \qquad (4.6)$$

where $\mu_m = x_m^* Ax_m$ and $x_m$ is normalized such that $x_m^* Bx_m = 1$.

The following will show that (4.6) is an equivalent form of (2.10). Since B is symmetric, it can be represented by a Cholesky factorization, ie. $B = TT^*$ where T is a triangular matrix. Letting $y_m = T^* x_m$, it follows that $(T^{-1})^* y_m = x_m$ and that $\mu_m = x_m^* A x_m = y_m^* T^{-1} A (T^{-1})^* y_m$ where $\mu_m$ is the Rayleigh quotient for the Hermitian matrix $T^{-1}A(T^{-1})^*$, calculated with the vector $y_m$. Substituting $x_m = (T^{-1})^* y_m$ and $B = TT^*$ into (4.6) results in the following equation

$$(A - TT^*(T^{-1})^* y_m y_m^* (T^{-1})A - \mu_m TT^*)((T^{-1})^* y_m + g_m) = -\mu_m TT^*(T^{-1})^* y_m$$

or

$$(A(T^{-1})^* - Ty_m y_m^*(T^{-1})A(T^{-1})^* - \mu_m T)(y_m + g_m) = -\mu_m T y_m$$

Premultiplying this equation by $T^{-1}$ yields

$$((T^{-1})A(T^{-1})^* - y_m y_m^*(T^{-1})A(T^{-1})^* - \mu_m I)(y_m + g_m) = -\mu_m y_m \qquad (4.7)$$

Equation (4.7) is of the same form as (2.10) except here the matrix is the Hermitian matrix $(T^{-1})A(T^{-1})^*$. Therefore, the successive solution of (4.6) will produce the eigenvalues and eigenvectors of the Generalized Eigenvalue Problem $Ax = \lambda Bx$ and the convergence of $x_m$ and $\mu_m$ for (4.6) is of the same nature as that of equation (4.7).

The corresponding solution of the generalized problem employing Rayleigh quotient inverse iteration would use the system

$$(A - \mu_m B)(x_m + g_m) = Bx_m \qquad (4.8)$$

Whatever advantage (2.10) would have over (2.1) for the algebraic problem (1.1) would hold in the comparison of (4.6) and (4.8), since $(A - \mu_m B)$ approaches a singular matrix as $\mu_m \to \lambda$, where $\lambda$ and its corresponding vector x solve $Ax = \lambda Bx$.

## V. CONDITIONING OF THE COEFFICIENT MATRIX

In Rayleigh quotient inverse iteration, one objection is that the coefficient matrix, $A - \mu_m I$, approaches a singular matrix as $\mu_m$ approaches an eigenvalue. This is aesthetically undesirable but generally does not cause computational problems because of the speed of convergence plus the fact that inaccuracies in any computer will prevent $A - \mu_m I$ from being represented exactly and thus prevent it from being exactly singular. However, the ill-conditioned nature of the coefficient matrix may present inaccuracies in the computation of the solution of (1.2) which is not desirable in either a theoretical or computational sense.

Let $H_m$ denote the coefficient matrix of the system of equations (2.10), i.e. $H_m = A - x_m x_m^* A - \mu_m I$. The possibility of $H_m$ becoming singular does exist and it is important to consider the conditions by which this may occur.

Let $\tilde{V} = (v_1, v_2, \ldots v_n)$ where $v_1, v_2, \ldots v_n$ are unit eigenvectors of A with corresponding eigenvalues of $\lambda_1, \lambda_2, \ldots \lambda_n$. Then

$$\tilde{V}^* Q_m \tilde{V} = \begin{pmatrix} (\lambda_1 - \mu_m) & & & \\ & (\lambda_2 - \mu_m) & & \\ & & \ddots & \\ & & & (\lambda_n - \mu_m) \end{pmatrix}$$

Let $x_m$ (where $||x_m|| = 1$) equal $v_j$ for some j such that $1 \leq j \leq n$. Then $\mu_m = \lambda_j$, and

$$\tilde{V}^* H_m \tilde{V} = \tilde{V}^* A\tilde{V} - \tilde{V}^* v_j v_j^* A\tilde{V} - \tilde{V}^* (\mu_m I)\tilde{V}$$

$$= \tilde{V}^* Q_m \tilde{V} - \tilde{V}^* v_j v_j^* A\tilde{V}$$

Since $\tilde{V}^* v_j v_j^* A\tilde{V} =$

$$\begin{pmatrix} 0 & & & & & \\ & \ddots & & & & \\ & & 0 & & \emptyset & \\ & & & \lambda_j & & \\ & & & & 0 & \\ & \emptyset & & & & \ddots \\ & & & & & & 0 \end{pmatrix}$$

then

$$\tilde{V}^* H_m \tilde{V} = \begin{pmatrix} (\lambda_1 - \mu_m) & & & & & & \\ & (\lambda_2 - \mu_m) & & & & & \\ & & \ddots & & & & \\ & & & (\lambda_{j-1} - \mu_m) & & & \\ & & & & -\mu_m & & \\ & & & & & (\lambda_{j+1} - \mu_m) & \\ & & & & & & \ddots \\ & & & & & & & (\lambda_n - \mu_m) \end{pmatrix}$$

Since $H_m$ is singular if $\tilde{V}^* H_m \tilde{V}$ is, then it should be observed that $\tilde{V}^* H_m \tilde{V}$ will be singular in the case that:

(1.) A has multiple eigenvalues

or (2.) A has at least one eigenvalue of 0.

This last situation can be easily eliminated by shifting the eigenbasis of A such that A has no zero eigenvalues. The eigenvectors, of course, would remain unchanged.

It should be noted here that the above analysis is considered from the standpoint of the $\lim_{m\to\infty}\{H_m\}$. It is, however, possible that $H_m$ for some m could become singular essentially randomly, or that $x_0$ could be chosen such that $H_0$ is immediately singular. This is unlikely in the first case and rare in the second.

## VI. GEOMETRY OF THE ITERATION, INCREMENTAL AND RESIDUAL VECTORS

The orthogonality relationship between $g_m$ and $x_m$ set forth in Chapter II was instrumental in the proof of convergence of the residual norm of the derived method. There are other geometries of the vectors $x_m$, $g_m$, and $r_m$ which are of interest.

For RQI, as $\mu_m \to \lambda$, from (1.2) it can be seen that $(A - \mu_m I)$ approaches a singular matrix and $||x_m + g_m|| \to \infty$. Successive iterations can therefore become somewhat ill-determined as $||r_m|| \to 0$. We would like to be able to state that convergence of $\{||r_m||\}$ to zero for the method of (2.10) implies that $g_m \to 0$. That this is the case for a certain class of matrices is proved below.

Theorem 6.1: Let A be an n x n Hermitian matrix with distinct eigenvalues. Let $\{x_m\}$ be the sequence of iteration vectors determined from the method of (2.10), and let $r_m$ be the corresponding residual vector. Then if $\{r_m\} \to 0$ as $m \to \infty$, $\{g_m\} \to 0$ as $m \to \infty$.

Proof: Suppose that $\lim_{m\to\infty} ||r_m|| = 0$ and $g_m \not\to 0$. Then there exists a convergent subsequence of vectors $\{x_k\} \subset \{x_m\}$, and an eigenvector z such that $\lim_{k\to\infty}\{x_k\} = z$, $\lim_{k\to\infty}||r_k|| = 0$, and $||g_k|| \neq 0$.

From the definition of $r_{m+\frac{1}{2}}$,

$$||r_{m+\frac{1}{2}}||^2 = \frac{||r_m||^2 + 2\mathrm{Re}\{g_m^* Q_m r_m\} + ||Q_m g_m||^2}{1 + ||g_m||^2}$$

and from (3.2),

$$||r_{m+\frac{1}{2}}||^2 \leq \frac{||g_m||^2 ||r_m||^2}{1 + ||g_m||^2}$$

These imply $2\mathrm{Re}\{g_k^* Q_k r_k\} + ||Q_k g_k||^2 \leq ||g_k||^2 ||r_k||^2 - ||r_k||^2$

Dividing this expression by $||g_k||^2$ (which does not approach zero) we obtain,

$$\frac{2\mathrm{Re}\{g_k^* Q_k r_k\}}{||g_k||^2} + \frac{||Q_k g_k||^2}{||g_k||^2} \leq ||r_k||^2 - \frac{||r_k||^2}{||g_k||^2}$$

Now

$$\lim_{k\to\infty} \frac{||2g_k^* Q_k r_k||}{||g_k||^2} \leq \lim_{k\to\infty} \frac{2||g_k|| \cdot ||Q_k|| \cdot ||r_k||}{||g_k||^2}$$

$$= \lim_{k\to\infty} \frac{2||Q_k|| \cdot ||r_k||}{||g_k||} = 0.$$

This implies that as $k \to \infty$, $||Q_k g_k||^2 \leq 0$ or $Q_k g_k \to 0$. Therefore $\lim_{k\to\infty}\{g_k\} = z$, an eigenvector corresponding to $\mu_k$. Since A is assumed to have distinct roots, this implies $x_k = \nu g_k$ as $k \to \infty$, ($\nu$ a constant), which is impossible since $g_k$ is constructed orthogonal to $x_k$. This is an obvious contradiction, and implies that if $\lim_{m\to\infty} ||r_m|| = 0$, then $||g_m|| \to 0$. QED

Hence for A, a matrix with distinct eigenvalues, the computation of successive vector iterates, $\{x_m\}$, for the method of (2.10) remains

quite well-determined even in the limit. Well-determined here, of course, means that $||x_m + g_m|| \not\to \infty$ as $m \to \infty$.

However, there is another sense by which $\{x_m\}$ may be considered to be well-determined. This is related to the fact that A is Hermitian and we will therefore expect orthogonal vectors. A starting vector $x_0$ is generally considered to be arbitrary but is not completely arbitrary in the sense that it is chosen from the space orthogonal to the set of all previously found eigenvectors of A. Let $V = v_1, v_2, \ldots v_k$, $k < n$, represent this set. No guarantee has yet been given that a vector chosen from V will after successive iterations (using (2.10)) remain orthogonal to V. If it does not then either it corresponds to a multiple root or it duplicates a vector already found as will be shown below. Let S represent the space orthogonal to all previously found eigenvectors.

Theorem 6.2: Let $x_m \subset S$ such that $\mu_m \neq \lambda_j$ for $j = 1, 2, \ldots k$. Then $x_{m+1}$ found by applying (2.10) is such that $x_{m+1} \subset S$.

Proof: Since $V = (v_1, v_2, \ldots v_k)$ then $V^* x_m = 0$. Let D denote the diagonal matrix with elements $\lambda_1 - \mu_m, \lambda_2 - \mu_m, \ldots, \lambda_k - \mu_m$. Further let L denote the diagonal matrix with elements $\lambda_1, \lambda_2, \ldots \lambda_k$. Recall that $H_m = A - x_m x_m^* A - \mu_m I$. Then,

$$V^* H_m = V^*(A - x_m x_m^* A - \mu_m I)$$

$$= V^* A - V^* x_m x_m^* A - \mu_m V^*$$

$$= LV^* - \mu_m V^* = DV^*$$

From (2.10), $H_m(x_m + g_m) = -\mu_m x_m$, which implies

$$V^* H_m(x_m + g_m) = -\mu_m V^* x_m$$

or $$DV^*(x_m + g_m) = 0.$$

Since $\mu_m \neq \lambda_j$ for $1 \leq j \leq k$, D is nonsingular and $D^{-1}$ exists. This implies

$$V^*(x_m + g_m) = 0$$

or $$V^* x_{m+1} = 0.$$

Hence $x_{m+1} \subset S$. QED

Consequently any vector iterate $x_m$ and therefore any subsequently determined eigenvector will remain orthogonal to all previously found eigenvectors. If an accepted vector does not correspond to a distinct root it may either duplicate a vector in V or be a vector corresponding to a multiple root in which case it might not be orthogonal to all vectors in V. Given then an eigenvalue which is very close to another previously found eigenvalue, then in order to determine which of the above cases holds, the vector z corresponding to the eigenvalue can be orthogonalized (using Gram-Schmidt process) with respect to all the vectors in V. If the result is the zero vector, then z is a duplicate vector and may be disregarded; otherwise, the orthogonalized vector z may be accepted as a vector corresponding to a legitimate multiple root. It has been found that the orthogonalization process may disturb z

somewhat and further iterations do not recover it without losing orthogonality again. However the loss in accuracy from the orthogonalization process has never been found to be too severe. In a completely analagous manner, it can be shown that Rayleigh quotient inverse iteration also preserves the orthogonality relationship between iteration vectors and previously found eigenvectors.

This loss of orthogonality for multiple roots can be eliminated completely if $g_m$ is chosen orthogonal not only to $x_m$ but also to all k previous eigenvectors, V. Following the derivation given in Chapter II, we have

$$P = (p_1, p_2, \ldots, p_{n-k-1}) \text{ and } (P, V, x_m)\begin{pmatrix} P^* \\ V^* \\ x_m^* \end{pmatrix} = I$$

with $$\begin{pmatrix} P^*A \\ V^* \\ x_m^* \end{pmatrix}(x_m + g_m) = \begin{pmatrix} \mu_m c \\ 0 \\ 1 \end{pmatrix}$$

Premultiplying by $(P, V, x_m)$ gives

$$(PP^*A + VV^* + x_m x_m^*)(x_m + g_m) = \mu_m g_m + x_m$$

$$\text{or} \quad (A - x_{\bar{m}} x_{\bar{m}}^* A - \mu_{\bar{m}} I - VV^*A)(x_{\bar{m}} + g_{\bar{m}}) = -\mu_{\bar{m}} x_{\bar{m}} \qquad (6.1)$$

The following is a proof that (6.1) will yield a vector $x_m + g_m$ which will be orthogonal to V, regardless of multiple roots.

Corollary 6.1: Let $x_m \subset S$ such that $\mu_m \neq 0$ for $j = 1, 2, \ldots, k$. Then $x_{m+1}$ found by applying (6.1) is such that $x_{m+1} \subset S$.

Proof: $V^*(A - x_m x_m^* A - \mu_m I - VV^* A) = V^* A - \mu_m V^* - V^* A$

$$= -\mu_m V^*$$

Premultiplying (6.1) by $V^*$ produces

$$-\mu_m V^*(x_m + g_m) = -\mu_m V^* x_m = 0$$

which since $\mu_m \neq 0$ implies $V^* x_{m+1} = 0$.

Therefore $x_{m+1} \subset S$. QED

Thus, except for the zero eigenvalue, successive application of (6.1) will never result in duplication. The problem of the zero eigenvalue can, as mentioned before, be resolved by shifting the eigenbasis such that A has no zero eigenvalues. The proof of convergence is immediate considering that $VV^* A(x_m + g_m) = 0$.

It should be noted that (6.1) is theoretically equivalent to deflating A after each eigenvector is determined and then applying equation (2.10), since $A - VV^* A = A - \lambda_1 v_1 v_1^* - \lambda_2 v_2 v_2^* - \ldots - \lambda_k v_k v_k^*$, and $x_m x_m^*(A - VV^* A) = x_m x_m^* A$.

There are several interesting and useful relationships between the iteration and residual vectors which can be expressed using inner

products. The following states and proves three of these relations.

Lemma 6.1: Let $x_m$, $x_{m+1}$, $r_m$, and $r_{m+\frac{1}{2}}$ be defined as above, and $\{x_m\}$ generated by equation (2.10). Then

(i) $x_{m+1}^* r_{m+\frac{1}{2}} = \mu_{m+1} - \mu_m$

(ii) $x_{m+1}^* r_m = \pm ||r_{m+\frac{1}{2}}||$

(iii) $x_m^* r_{m+\frac{1}{2}} = x_{m+1}^* r_m$

Proof: (i) $$x_{m+1}^* r_{m+\frac{1}{2}} = \left(\frac{x_m^* + g_m^*}{||x_m + g_m||}\right)\left(\frac{(A - \mu_m I)(x_m + g_m)}{||x_m + g_m||}\right)$$

$$= \frac{(x_m^* + g_m^*)A(x_m + g_m) - \mu_m(1 + g_m^* g_m)}{||x_m + g_m||^2}$$

$$= \Gamma_{m+1} - \Gamma_m$$

(ii) $$x_{m+1}^* r_m = \frac{(x_m^* + g_m^*)(A - \mu_m I)x_m}{||x_m + g_m||} = \frac{x_m^* A x_m - \mu_m + g_m^* A x_m}{||x_m + g_m||}$$

$$= \frac{g_m^* A x_m}{||x_m + g_m||} = \frac{g_m^* r_m}{||x_m + g_m||} \qquad (6.2)$$

Consider RQI as given by (1.2) and (2.1), where $g_m$ is not

necessarily orthogonal to $x_m$. Equation (2.1) implies that

$$(A - \mu_m I)\frac{(x_m + g_m)}{||x_m + g_m||} = \frac{x_m}{||x_m + g_m||}$$

or $(A - \mu_m I)x_{m+1} = kx_m$ where $k = \dfrac{1}{||x_m + g_m||}$.

Thus k is real and positive. Premultiplying (1.2) by $x_m$, we see that

$$k = x_m^*(A - \mu_m I)x_{m+1} = x_{m+1}^*(A - \mu_m I)x_m = x_{m+1}^* r_m.$$

Therefore $x_{m+1}^* r_m$ (for RQI) is real and positive. Recall in Chapter IV that the normalized vector $x_{m+1}$ found from (2.10) was found to be plus or minus the corresponding normalized vector found from (1.2). The residual vector $r_m = (A - \mu_m I)x_m$ will equal its corresponding residual vector for RQI if the corresponding normalized iteration vectors at the $m^{th}$ step, $x_m$, were the same. If they differed by a sign, then so would the residual vectors. Thus, in general,

$$x_{m+1}^* r_{m[2.10]} = \pm x_{m+1}^* r_{m[1.2]}$$

and $x_{m+1}^* r_{m[2.10]}$ is real but not necessarily positive, and therefore $g_m^* r_m$ in (6.2) is real.

We know from the convergence proof given in Chapter III, that

$$\frac{||g_m^* r_m||^2}{||x_m + g_m||^2} = ||r_{m+\frac{1}{2}}||^2$$

Since $g_m^* r_m$ is real, $(g_m^* r_m)^2 = ||g_m^* r_m||^2$, and therefore

$$\frac{g_m^* r_m}{||x_m + g_m||} = \pm ||r_{m+\frac{1}{2}}||$$

Combining this with the result of (6.2) implies

$$x_{m+1}^* r_m = \pm ||r_{m+\frac{1}{2}}||$$

(iii) $$x_m^* r_{m+\frac{1}{2}} = x_m^*(A - \mu_m I)x_{m+1} = \frac{x_m^*(A - \mu_m I)(x_m + g_m)}{||x_m + g_m||}$$

Also, $$x_{m+1}^* r_m = \frac{(x_m^* + g_m^*)(A - \mu_m I)x_m}{||x_m + g_m||} = \frac{x_m^*(A - \mu_m I)(x_m + g_m)}{||x_m + g_m||}$$

since $x_{m+1}^* r_m$ is real. Therefore $x_m^* r_{m+\frac{1}{2}} = x_{m+1}^* r_m$. QED

Since $x_{m+1}^* r_m$ is real, it can be expressed as

$$x_{m+1}^* r_m = ||r_m|| \cos\gamma_m \qquad (6.3)$$

where $\gamma_m$ is the "angle" between $x_{m+1}$ and $r_m$. Then

$$r_{m+1} = (A - \mu_{m+1} I)x_{m+1}$$

$$= (A - \mu_m I)x_{m+1} - (\mu_{m+1} - \mu_m)x_{m+1}$$

$$= r_{m+\frac{1}{2}} - (\mu_{m+1} - \mu_m)x_{m+1}$$

which implies that

$$||r_{m+1}||^2 = ||r_{m+\frac{1}{2}}||^2 - 2(\mu_{m+1} - \mu_m)x^*_{m+1}r_{m+\frac{1}{2}} + (\mu_{m+1} - \mu_m)^2$$

$$= (x^*_{m+1}r_m)^2 - (\mu_{m+1} - \mu_m)^2 \quad \{\text{from (i) and (ii), Lemma (6.1)}\}$$

$$= ||r_m||^2\cos^2\gamma_m - (\mu_{m+1} - \mu_m)^2 \quad \{\text{from (6.3)}\} \quad (6.4)$$

This latter result can be seen in Parlett (6) and Wilkinson (9), where it is proved for RQI in an equivalent manner without using the notation of $r_{m+\frac{1}{2}}$. Notice that the above is another way of expressing the identity (2.2) and that identities (2.2) and (6.4) hold for both Rayleigh quotient inverse iteration and the method of (2.10).

We proceed now with the statement and proof of an interesting result which establishes another orthogonality relationship between certain residual vectors.

Lemma 6.2: The residual vector $r_{m+\frac{1}{2}}$ is parallel to the iteration vector $x_m$ and is orthogonal to the residual vector $r_m$.

Proof: From (ii) and (iii) of Lemma (6.1), it is clear that

$$x^*_m r_{m+\frac{1}{2}} = \pm\, ||r_{m+\frac{1}{2}}|| \quad (6.5)$$

Since $x^*_m r_{m+\frac{1}{2}}$ is real, then from the cosine formula,

$$x^*_m r_{m+\frac{1}{2}} = ||r_{m+\frac{1}{2}}||\cos\eta_m \quad (6.6)$$

where $\eta_m$ is the "angle" between $x_m$ and $r_{m+\frac{1}{2}}$. Equations (6.5) and (6.6) imply therefore that if $||r_{m+\frac{1}{2}}|| \neq 0$, then $\cos\eta_m = \pm 1$, which in turn implies that $x_m$ and $r_{m+\frac{1}{2}}$ are parallel.

Since $r_m = (A - \mu_m I)x_m$, then $x_m^* r_m = x_m^* A x_m - \mu_m = 0$, and $x_m$ and $r_m$ are orthogonal. Therefore $r_{m+\frac{1}{2}}$, which is parallel to $x_m$ is also orthogonal to $r_m$. QED

Since $$r_{m+\frac{1}{2}}^* r_m = \frac{(x_m^* + g_m^*)Q_m^2 x_m}{||x_m + g_m||} = 0,$$ this implies that

$$x_m^* Q_m^2 x_m + g_m^* Q_m^2 x_m = 0.$$

The fact that $||r_m||^2 = ||Q_m x_m||^2 = x_m^* Q_m^2 x_m$ means that $g_m^* Q_m^2 x_m$ is real and moreover equal to $-||r_m||^2$. Thus $2\mathrm{Re}\{g_m^* Q_m r_m\} = 2 g_m^* Q_m r_m$ (in Theorem 3.1 and Theorem 6.1) since $g_m^* Q_m r_m$ is real. This somewhat simplifies the expression for $||r_{m+\frac{1}{2}}||^2$ since it can now be written

$$||r_{m+\frac{1}{2}}||^2 = \frac{-||r_m||^2 + ||Q_m g_m||^2}{1 + ||g_m||^2}.$$

## VII. SOLUTION OF THE SYSTEM OF EQUATIONS (2.10)

The system of linear equations can, of course, easily be solved using Gaussian elimination with full pivoting for maximum stability. However, this is done at some sacrifice of speed. The number of multiplications and additions for Gaussian elimination is of the order of $n^3/3$. The coefficient matrix $A - x_m x_m^* A - \mu_m I$ is a full matrix and is not Hermitian. However, a method has been devised (using no pivoting) which shows a drastic improvement in speed.

This speedup will be realized by tridiagonalizing A at the outset, using tridiagonal A from then on, and finally back transforming the eigenvectors determined since they will be the eigenvectors of tridiagonal A instead of full A. If A is tridiagonalized, this does not mean that $H_m$ will be tridiagonal. However, due to the fact that $x_m x_m^* A$ is a simple product matrix, $H_m$ will assume a particular form. Although a full matrix, $H_m$ will be such that its $i^{th}$ and $i+1^{st}$ rows have respectively the forms

$$i^{th} \text{ row } \; -x_i\bar{t}_1,\ -x_i\bar{t}_2,\ \ldots,\ -x_i\bar{t}_{i-2},\ a_{i,i-1}-x_i\bar{t}_{i-1},\ a_{ii}-x_i\bar{t}_i-\mu_m,$$

$$a_{i,i+1}-x_i\bar{t}_{i+1},\ -x_i\bar{t}_{i+1},\ \ldots,\ -x_i\bar{t}_n$$

$$i+1^{st} \text{ row } \; -x_{i+1}\bar{t}_1,\ -x_{i+1}\bar{t}_2,\ \ldots,\ -x_{i+1}\bar{t}_{i-1},\ a_{i+1,i}-x_{i+1}\bar{t}_i,$$

$$a_{i+1,i+1}-x_{i+1}\bar{t}_{i+1}-\mu_m,\ a_{i+1,i+2}-x_{i+1}\bar{t}_{i+2},\ -x_{i+1}\bar{t}_{i+2},\ \ldots,$$

$$-x_{i+1}\bar{t}_n$$

where the vector $t^* = x_m^* A$.

Consider $p_{i-1} = -x_i/x_{i-1}$. Starting with the $n-1^{th}$ and $n^{th}$ rows and proceeding upwards in groups of two, the $i^{th}$ row will be multiplied by $p_i$ and added to the $i+1^{st}$ row, producing for these 2 rows:

$i^{th}$ row $\quad 0, \ldots, 0, p_{i-1}a_{i-1,i-2}, p_{i-1}(a_{i-1,i-1}-\mu_m)+a_{i,i-1},$

$$p_{i-1}a_{i-1,i}+(a_{i,i}-\mu_m),\ a_{i,i+1},\ 0, \ldots, 0$$

$i+1^{st}$ row $\quad 0, \ldots, 0, p_i a_{i,i-1}, p_i(a_{i,i}-\mu_m)+a_{i+1,i},$

$$p_i a_{i,i+1}+(a_{i+1,i+1}-\mu_m),\ 0, \ldots, 0$$

Thus the form of (2.10) (with x's representing nonzero terms) would be

$$\begin{pmatrix} XXXX \ldots X \\ XXX \\ XXXX \\ \ XXXX \\ \ldots\ \ \emptyset \\ \ \ \ldots \\ \ \ \ldots \\ \ \ \ldots \\ \ \ \ldots \\ \emptyset\ \ \ldots \\ \ \ \ldots \\ \ \ \ XXXX \\ \ \ \ XXXX \\ \ \ \ \ XXXX \\ \ \ \ \ XXX \end{pmatrix} (x_m + g_m) = \begin{pmatrix} X \\ 0 \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ 0 \end{pmatrix} \qquad (7.1)$$

The next step will involve the elimination of the sub-sub diagonal elements, beginning with the third row and proceeding to the $n^{th}$, producing the form

$$\begin{pmatrix} XXX & \cdots & X \\ XXX & & \\ & XXX & \\ & & XXX \\ & & \vdots \\ & & \vdots \\ & & \vdots \\ & & \vdots \\ \emptyset & & \vdots \\ & & XXX \\ & & XXX \\ & & XX \end{pmatrix} (x_m + g_m) = \begin{pmatrix} X \\ 0 \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ 0 \end{pmatrix} \qquad (7.2)$$

The third step consists of eliminating the first through $(n-1)^{st}$ elements in the top row. The pivot element for the first element of the first row will be the first element of the second row; the pivot element to eliminate the second element of the first row will be the second element of the third row, and so on. The right side again will remain unchanged and produce the following form

$$\begin{pmatrix} 000 & \cdots & X \\ XXX & & \\ & XXX & \\ & & XXX \\ & & \vdots \\ & & \vdots \\ & & \vdots \\ & & \vdots \\ \emptyset & & \vdots \\ & & XXX \\ & & XXX \\ & & XX \end{pmatrix} (x_m + g_m) = \begin{pmatrix} X \\ 0 \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ 0 \end{pmatrix} \qquad (7.3)$$

Conceptually the next step would be to move the second through $n^{th}$ rows up one and the first row to the position of the $n^{th}$ row. This, of course, would not actually be done computationally. The form of the system (2.10) would then be

$$\begin{pmatrix} XXX & & & & \\ XXX & & & & \\ XXX & & & & \\ \cdots & & & & \\ \cdots & & & & \\ \cdots & & & & \\ \cdots & & & & \\ & XXX & & & \\ & XXX & & & \\ & XX & & & \\ & X & & & \end{pmatrix} (x_m + g_m) = \begin{pmatrix} 0 \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ 0 \\ X \end{pmatrix} \qquad (7.4)$$

The fifth and final step is that of back substitution in the tridiagonal matrix of (7.4).

The following table gives the number of divisions, multiplications, and additions that the above procedure would require.

TABLE 7.1 Computations for Linear System Solver

| Step | Divisions | Multiplications | Additions(subtractions) |
|---|---|---|---|
| 7.1 | n - 1 | 4n - 1 | 3n |
| 7.2 | n - 2 | 2(n - 2) | 2(n - 2) |
| 7.3 | n - 1 | 2(n - 1) | 2(n - 1) |
| 7.4 | n | 2n - 3 | n - 1 |
| total | 4n - 4 | 10n - 10 | 8n - 7 |

Thus the number of multiplications is of the order n as opposed to $n^3/3$ which for larger matrices can result in significant savings.

There are other sections of code for the method which are performed every iteration but do not directly involve solving the system of equations in (2.10). These are given in Table 7.2 with, again, the assumption that A is tridiagonal.

TABLE 7.2 Computations for Main Body of Code

| Computation | Divisions | Multiplications | Additions(subtractions) |
|---|---|---|---|
| $x_m^* A = t^*$ | 0 | 3n - 2 | 3n - 2 |
| $(x_m^* A)x_m = \mu_m$ | 0 | n | n |
| $x_m^* x_m$ | 0 | n | n |
| Normalize $x_m + g_m$ | n | 0 | 0 |
| $r_m$ | 0 | n | n |
| $\|\|r_m\|\|^2$ | 0 | n | n |
| total | n | 7n - 2 | 7n - 2 |

The total number of computations per iteration then would be 5n - 4 divisions, 17n - 12 multiplications, and 15n - 9 additions or subtractions.

Taking advantage of the fact that there is no pivoting, some of the above operations can be combined resulting in a slight decrease in time resulting from trading divisions for multiplications.

The program actually written to solve (2.10) computes essentially (upon back substitution)

$$\tilde{x}_i = p_{i+1} \cdot \tilde{x}_{i+1} + q_{i+2} \cdot \tilde{x}_{i+2} \qquad (7.6)$$

with $$p_i = (x_{i-1} \cdot (d_i - \mu_m) - x_i \cdot (e_i + e_{i-1} \cdot q_i))/h_i$$

$$q_i = (e_i \cdot x_{i-2})/h_{i-1}$$

$$h_i = x_i \cdot (d_{i-1} - \mu_m + e_{i-1} \cdot p_{i-1}) - e_i \cdot x_{i-1}$$

where $d_i$ is the $i^{th}$ diagonal element, $e_i$ is the $i^{th}$ super(sub) diagonal element, $x_i$ is the $i^{th}$ component of the vector x at the $m^{th}$ iteration, and $\tilde{x}_i$ is the $i^{th}$ component of the vector x at the $(m+1)^{st}$ iteration. There are several other calculations to get rid of the top row, etc. The complete program can be seen in Appendix B.

The method of solving (2.10) outlined above generally works very well and is of course quite fast. It can begin to break down, however, whenever one or more of the elements of the vector x are converging to 0. This is due, of course, to the pivot elements $p_{i-1}=-x_i/x_{i-1}$. Even though (7.6) has eliminated the actual division by performing the computation in a different way, the effect of the division by a small number is still felt because the computation now is multiplying by a very small number. The result will be that the residual norm will be reduced to a point and then will stop decreasing

even though it remains above the accepted tolerance. In such a case Gaussian elimination could be used or the small element of the vector could be perturbed somewhat.

Several other savings have been realized such that the program which appears in Appendix B actually calculates the solution of the system with approximately 2n divisions, 10n multiplications, and 9n additions, essentially eliminating half of the divisions in TABLE 7.1.

## VIII. A STATISTICAL STUDY OF THE ITERATION PROCESS

In almost any iterative process, the number of computations (additions, multiplications, etc.) can be calculated per iteration. What is not generally known is the number of iterations. Naturally any good estimate of the number of iterations required, whether in the form of an upper bound or average, would be of great help in establishing a correspondingly reasonable estimate of the total computations. Aside from the general interest of such an estimate, it would enable a meaningful comparison to be made between the method in question and other similar methods, whether iterative or non-iterative.

The only other alternative to this type of comparison would be to examine the corresponding execution times of various methods. This is highly unsatisfactory due to the inaccuracies in computer clocks and varying hardware between computers. Also this method of comparison does not allow representation of computations according to the order of the matrix and the benefit of extrapolation as the order increases.

Considering method (2.10), it would then be desirable to know the average number of iterations per vector, given a matrix of order n. A statistical test using regression has been performed to try to attempt an estimate of any correlation between order and number of iterations. Let the variable X represent a class of matrix of a certain order, and Y represent the average number of iterations per

vector for the matrix class X. The experiment was performed with orders five through fifty with X increasing by five at every step. Thus there are ten data points and with the assumption of normality within classes, we shall assume that the correlation coefficient is distributed according to the Student's-t distribution. Within each order class, ie. for every $X_i$, there are 15 matrices. The total number of iterations does not include the last vector since theoretically no iterations are performed for it. In actual practice the final vector takes usually one iteration or none. This is significantly different from the number of iterations of the other vectors and thus has been discarded. Each value of Y then is calculated by summing the total number of iterations for the 15 matrices (not counting those iterations of the $n^{th}$ vector), and dividing by 15(n - 1). The matrices themselves have been randomly generated with elements between 0 and 1. They are, of course, symmetric.

The following table presents the calculated values of Y and the relevant values of $X^2$, $Y^2$, and XY needed in the regression analysis.

TABLE 8.1 Regression Data for test of Matrix Orders

| X | Y | $X^2$ | $Y^2$ | XY | |
|---|---|---|---|---|---|
| 5 | 4.51667 | 25 | 20.40031 | 22.58335 | |
| 10 | 5.17037 | 100 | 26.73273 | 51.70370 | |
| 15 | 5.41904 | 225 | 29.36599 | 81.28560 | |
| 20 | 5.61654 | 400 | 31.54552 | 112.33038 | |
| 25 | 5.95000 | 625 | 35.40250 | 148.75000 | |
| 30 | 6.15862 | 900 | 37.92860 | 184.75860 | |
| 35 | 6.28235 | 1225 | 39.46792 | 219.88225 | |
| 40 | 6.52568 | 1600 | 42.58450 | 261.02720 | |
| 45 | 6.68333 | 2025 | 44.66690 | 300.74985 | |
| 50 | 6.77959 | 2500 | 45.96284 | 338.97950 | |
| 275 | 59.10219 | 9625 | 354.05781 | 1722.05085 | sum |

From TABLE 8.1, the following regression calculations can be made:

$$\sum x^2 = \sum (X - \bar{X})^2 = \sum X^2 - (\sum X)^2/n = 2062.50000$$

$$\sum y^2 = \sum (Y - \bar{Y})^2 = \sum Y^2 - (\sum Y)^2/n = 4.75092$$

$$\sum xy = \sum (X - \bar{X})(Y - \bar{Y}) = \sum XY - (\sum X)(\sum Y)/n = 96.74063$$

$$\sum d_{y \cdot x}^2 = \sum y^2 - \{(\sum xy)^2/\sum x^2\} = .21334$$ where the $d_{y \cdot x}$'s

are the deviations from regression, ie. $d_{y \cdot x} = Y - \hat{Y}$, where

$\hat{Y} = \bar{Y} + b(X - \bar{X})$ with b the regression coefficient. Then $S^2_{y \cdot x} = \sum d^2_{y \cdot x}/(n - 2) = .02667$ where $S^2_{y \cdot x}$ is the mean square deviation from regression. $S_{y \cdot x} = .16334$ or sample standard deviation from regression.

Then the sample standard deviation of the regression coefficient is given by $S_b$ where

$$S_b = S_{y \cdot x}/\sqrt{\sum x^2} = .00359$$

and the regression coefficient is given by b where

$$b = \sum xy/\sum x^2 = .04690$$

with $n - 2 = 8$ degrees of freedom.

A test of significance of the regression coefficient b is given by $t = b/S_b = 13.06406$ (and 8 d.f.) which is statistically significant for a 99.9% confidence interval.

Therefore the regression line is

$$\hat{Y} - \bar{Y} = b(X - \bar{X})$$

or $$\hat{Y} = .0469X + 4.62047 \qquad (8.1)$$

Although this straight line exhibits only about a 4.7% rise, the correlation is strong between the number of iterations per vector and the order of the matrix. These results hold for orders five through fifty and it seems unlikely that a major unpredicted change would

occur after order fifty.

Equation (8.1) can now be used to estimate the total number of computations of a kind, say multiplications, given the order of the matrix. Recall in Chapter VII, it was stated that the program in Appendix B calculates the solution of the system (2.10) with 10n - 10 multiplications. In addition, computed at every iteration are the variables in TABLE 7.2, for a total of 7n - 2 multiplications.

Also entering the calculations are the multiplications required to form new starting vectors. Recall this can be computed by selecting any column of the matrix $I - \sum_{i=1}^{k} v_i v_i^*$ where $v_k$ was the last vector found. This requires then approximately $n^2 - n$ multiplications for the entire problem.

Combining the above with the average number of iterations per vector found in (8.1), the total number of multiplications for the problem is approximately

$$(17n - 12)(.0469n + 4.62047)(n - 1) + n^2 - n$$

$$\text{or} \quad .7973n^3 + 78.1897n^2 - 134.4308n + 55.4456 \qquad (8.2)$$

The total number of iterations for the popular QR method has been determined by Businger (2) to be approximately $4n^3$. Figure 8.1 represents the differences between the two methods.

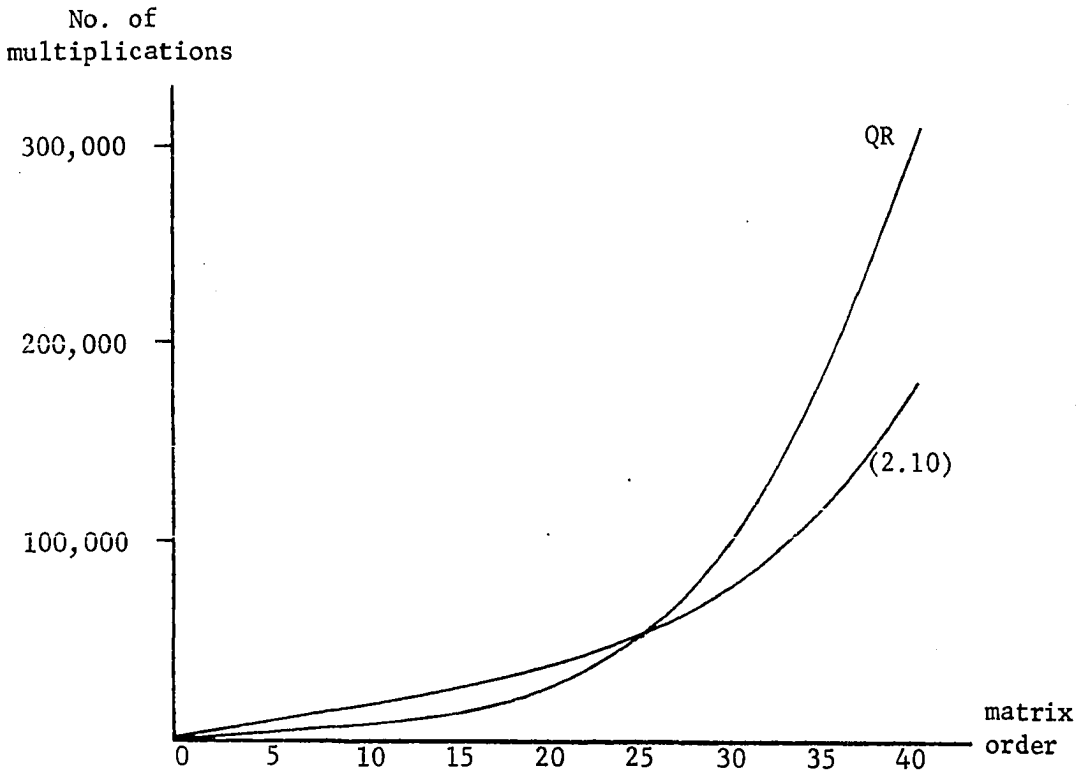


FIGURE 8.1 Crossover Point between QR method and (2.10)

The two methods thus have an effective cross-over point of about order 25. Therefore above order 25, it will be profitable to use method (2.10) instead of QR. The above graphs are only for multiplications, of course, but since multiplications are the predominant computation in both methods, the cross-over point for all computations (additions, multiplications, divisions) is also close to 25.

Of course, RQI which requires less work in the solution of the system of equations, would have an even lower cross-over point with the QR method.

This points up the fact that after many years of peripheral use, inverse iteration is becoming more popular as a general method for the solution of the Algebraic Eigenvalue Problem.

Of further interest in many iterative methods is the part diagonal dominance plays. For method (2.10) diagonal dominance is irrelevant as far as convergence itself is concerned but it would be interesting to determine whether diagonal dominance affected speed of convergence. Another regression test was run to try to determine this effect, if any.

Random matrices with elements between 0 and 1 were again generated for matrices only of order 10. Each matrix was then made minimally diagonally dominant, that is, each element $a_{ii}$ was set equal to $\sum_{\substack{j=1 \\ j\neq i}}^{n} a_{ij}$. Again ten classes were set up. Here each class was constructed so that its diagonal elements, after calculated as above, were multiplied by X, where X was incremented by 2 at every class. TABLE 8.2 gives the relevant data. Here again, $Y_i$ represents the average number of iterations per vector for the class $X_i$, and each class contains 15 matrices.

TABLE 8.2 Regression Data for test of effect of diagonal dominance

| X | Y | $X^2$ | $Y^2$ | XY |
|---|---|---|---|---|
| 1 | 4.97777 | 1 | 24.77819 | 4.9777 |
| 3 | 4.89629 | 9 | 23.97366 | 14.68887 |
| 5 | 4.96296 | 25 | 24.63097 | 24.81480 |
| 7 | 5.05185 | 49 | 25.52119 | 35.36295 |
| 9 | 5.02222 | 81 | 25.22269 | 45.19998 |
| 11 | 5.01481 | 121 | 25.14832 | 55.16291 |
| 13 | 4.98518 | 169 | 24.85202 | 64.80734 |
| 15 | 4.97777 | 225 | 24.77819 | 74.66655 |
| 17 | 5.00740 | 289 | 25.07405 | 85.12580 |
| 19 | 4.91111 | 361 | 24.11900 | 93.31109 |
| 100 | 49.80736 | 1330 | 248.09828 | 498.11806 |

Using the same analysis as before,

$$\sum x^2 = 330, \quad \sum y^2 = .02097, \quad \sum xy = .04446,$$

$$\sum d^2_{y \cdot x} = .02096, \quad s^2_{y \cdot x} = .00262, \quad s_{y \cdot x} = .05119,$$

$s_b = .00281$, $b = .00012$, and $t = .04270$ which is not significant. Thus the null hypothesis can not be rejected. Although not a strong statement, we can consider that at least for order 10 matrices there is no appreciable indication of any effect caused by the increasingly diagonally dominant characteristic of the matrices.

# IX. CONCLUSION

A numerical method has been derived for the solution of the Algebraic Eigenvalue Problem for Hermitian matrices. The method has been shown to exhibit global convergence and is equivalent to Rayleigh quotient inverse iteration when the corresponding coefficient matrices are nonsingular. The coefficient matrix of the method remains, in general, nonsingular for iterations on distinct eigenvalues. An algorithm has been presented to implement the method.

In addition extensions have been derived - one to eliminate the problem of duplication on multiple roots by the addition of the term $-VV^*A$, and the other an extension of the method to the solution of the Generalized Eigenvalue Problem $Ax = \lambda Bx$ where A and B are Hermitian, and B is positive definite.

A method was devised to enable the system of linear equations to be solved in $10n$ multiplications instead of $n^3/3$ as in Gaussian elimination. In addition a statistical study was made establishing a slight correlation between average number of iterations per vector and the order of the matrix.

It is worth noting again the interesting fact that the derivation was accomplished by taking a critical point of the numerator of the difference of $\mu_{m+1}$ and $\mu_m$. Also worth noting is the fact that the basis set up was then eliminated and did not appear in the final linear system.

There are, however, several questions which remain unanswered. One of these lies in the convergence proof. A condition for convergence required $H_m^{-1}$ to exist for all m, and although at the limit as $m \to \infty$, we know that $H_m^{-1}$ will exist except for multiple roots, nevertheless it is possible that $H_m$ might become nonsingular somewhere before the limit. This would be extremely rare and experimental evidence to date indicates that it occurs when RQI oscillates, which is also rare. Naturally, it would be highly desirable to show that given $H_0$ nonsingular, $H_m$ is nonsingular for all m, except possibly at the limit (for multiple roots).

Of great interest also is the fact that method (2.10) and by equivalence RQI can be derived by taking a critical point of the numerator of $\mu_{m+1} - \mu_m$. It is not known however whether this critical point is a maximum, minimum, or saddle point. There is no reason to expect (since we are dealing with a signed term $\mu_{m+1} - \mu_m$) that the critical point is always a maximum, or always a minimum, etc. And there is no useful criterion to judge under what conditions the critical point is a maximum, minimum, or saddle point.

Finally, as in all iterative methods of this type, a good initial approximation may drastically reduce the number of iterations required. The method of choosing starting vectors discussed before does little more than create a starting vector which is orthogonal to all the previous vectors. It has been shown that this vector will generally

remain orthogonal to the previously found eigenvectors. Only for the $n^{th}$ vector does this procedure yield an excellent starting vector. The $n^{th}$ vector when orthogonalized and normalized should theoretically be the eigenvector. In actual fact it takes 0 or 1 iterations. This procedure of starting vectors does not contribute to the speed of convergence for vectors 1 through n-1.

## X. REFERENCES


1. Bodewig, E. Matrix Calculus. Amsterdam: North-Holland, 1959.

2. Businger, P.A. "SSEV and HSEV - Selected Eigenvalues of Real Symmetric and Complex Hermitian Matrices," Computing Science Technical Report #4, Bell Laboratories, Murray Hill, New Jersey, 1972.

3. Chuang, P. "Description and Error Analysis of a Rapidly Converging Norm Reduction Method for Eigenvectors." Unpublished Ph.D. dissertation, Iowa State University, Ames, 1970.

4. Fox, L. An Introduction to Numerical Linear Algebra. New York: Oxford University Press, 1965.

5. Ostrowski, A.M. "On the Convergence of the Rayleigh Quotient Iteration," Arch. Rat. Mech. Anal., 1958-59.

6. Parlett, B.N.; and Kahan W. "On the Convergence of a Practical QR Algorithm," IFIP 68, Amsterdam: North-Holland, 1969.

7. Peters, G.; and Wilkinson, J.H. "The Calculation of Specified Eigenvectors by Inverse Iteration." Linear Algebra. Edited by J. Wilkinson and C. Reinsch. New York: Springer-Verlag, 1971.

8. Wilkinson, J.H. The Algebraic Eigenvalue Problem. Oxford: Clarendon Press, 1965.

9. Wilkinson, J.H. "Inverse Iteration in Theory and in Practice," Istituto Nazionale di Alta Matematica, Symposia Mathematica, Volume X, 1972.

10. Wilkinson, J.H. Rounding Errors in Algebraic Processes. Englewood Cliffs, N.J.: Prentice-Hall, Inc., 1963.

## XI. ACKNOWLEDGEMENTS

The author is especially indebted to Dr. Robert J. Lambert for his help and guidance during graduate school and during the preparation of this thesis.

## XII. APPENDIX A: IMPLEMENTATION OF THE METHOD

This appendix includes a Fortran implementation of the method outlined on pages 8-9. The matrix read in is S and is tridiagonalized by means of the Eispack subroutine TRED1. D contains the resultant diagonal, and E contains the sub-diagonal. TRBAK1 is another Eispack subroutine which backtransforms the vectors so that they become eigenvectors of the full matrix. Eispack is a series of programs distributed by Argonne National Laboratory.

```
      SUBROUTINE RSC(L,N,S,W,Z,CNORM,THRES1)
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*8 S(L,L),W(L),Z(L,L),CNORM(L),X(10),TEMP(10),MACHEP,
     * D(10),E(10),E2(10)
      LOGICAL DOUBLE, RANDOM, OSC, RECOMP
      DATA MACHEP/Z3410000000000000/
      CALL ERRSET(207,256,0,0,0,0)
      FIRST=1.D0/DSQRT(DFLOAT(N))

C     TRIDIAGONALIZE MATRIX

      CALL TRED1(L,N,S,D,E,E2)
      ANORM=0.D0
      E(N+1)=0.D0
      RECOMP=.FALSE.
      GO TO 742
  741 RECOMP=.TRUE.

C     COMPUTE NORM OF MATRIX

  742 DO 25 I=1,N
      TEMP1=DABS(E(I))+DABS(D(I))+DABS(E(I+1))
   25 IF(TEMP1.GT.ANORM) ANORM=TEMP1
      IF(RECOMP) GO TO 744
      KC=2.2D0*ANORM
      DO 148 I=1,N
      Z(I,1)=0.D0
      Z(I,N)=0.D0
      X(I)=FIRST
```

```
C     SHIFT BASIS TO ELIMINATE ZERO EIGENVALUE

 148  D(I)=D(I)+KC
      GO TO 741

C     COMPUTE THRESHOLD TOLERANCE

 744  THRES1=ANORM*N*MACHEP*40.D0
      IX = 479451537
      PREVNM=1.D70
      OSC=.FALSE.
      INTL=1
      DOUBLE=.FALSE.
      RANDOM=.FALSE.
      M=0
      GO TO 986
 637  RANDOM=.FALSE.
      PREVNM=1.D70
      M=0
 15   XNORM=0.D0

C     COMPUTE NORM OF VECTOR

      DO 16 I=1,N
 16   XNORM=XNORM+X(I)**2
      XNORM=1.D0/DSQRT(XNORM)

C     TEST FOR ZERO VECTOR

      IF(XNORM.LT.1.D7) GO TO 666
      DOUBLE=.FALSE.
      WRITE(6,232)
 232  FORMAT(' ','ZERO VECTOR')
 703  RANDOM=.TRUE.
      GO TO 777

C     NORMALIZE VECTOR TO LENGTH ONE

 666  DO 17 I=1,N
 17   X(I)=X(I)*XNORM

C     TEMP= MATRIX A TIMES VECTOR X; XLAM= RAYLEIGH QUOTIENT

 986  TEMP(1)=D(1)*X(1)+E(2)*X(2)
      C(N)=E(N)*X(N-1)
      TEMP(N)=C(N)+D(N)*X(N)
      IF(.NOT.OSC) SAVXLM=XLAM
      XLAM=X(1)*TEMP(1)+X(N)*TEMP(N)
```

```
      DO 21 I=2,NM1
      C(I)=E(I)*X(I-1)
      TEMP(I)=C(I)+D(I)*X(I)+E(I+1)*X(I+1)
 21   XLAM=XLAM+X(I)*TEMP(I)
C
C     COMPUTE MODIFIED RESIDUAL NORM

      RNORM=0.D0
      DO 28 I=1,N
 28   RNORM=RNORM+DABS(TEMP(I)-XLAM*X(I))
      IF(OSC) GO TO 804

C     TEST FOR DOUBLE ROOT

      IF(DOUBLE) GO TO 750

C     DEFAULT STOP IF MORE THAN 30 ITERATIONS

      IF(RNORM.LE.THRES1.OR.M.GT.30) GO TO 70
      IF(RNORM.LT.PREVNM) GO TO 804

C     TEST FOR OSCILLATION

      IF(RNORM.GT.DSQRT(THRES1)) GO TO 703

C     OSCILLATION

      OSC=.TRUE.
      WRITE(6,222)
 222  FORMAT(' ','OSC')
      GO TO 777
 804  PREVNM=RNORM

C     SOLVE SYSTEM OF EQUATIONS (2.10)

      CALL SOLVE(D,E,X,SAVXLM,N,L,TEMP,OSC)
      M=M+1
      GO TO 15
 70   IF(INTL.EQ.1) GO TO 750

C     CHECK FOR MULTIPLE ROOTS

 71   DO 699 I=1,II
      IF(DABS(XLAM-W(I))/(DABS(XLAM)+DABS(W(I))).LE..1D-10) GO TO 999
 699  CONTINUE
 750  W(INTL)=XLAM
```

```
      CNORM(INTL)=RNORM
      DO 500 J=1,N
  500 Z(J,INTL)=X(J)
  488 IF(INTL.EQ.N) GO TO 7165
      INTL=INTL+1
 1001 DOUBLE=.FALSE.
      II=INTL-1

C     COMPUTE NEW STARTING VECTOR

      SUM=0.D0
      DO 2002 I=1,N
      X(I)=FIRST
 2002 SUM=SUM+Z(I,II)
      ZMPL=SUM*FIRST
      RANDOM=.TRUE.
      DO 690 I=1,N
      Z(I,N)=Z(I,N)+ZMPL*Z(I,II)
  690 X(I)=X(I)-Z(I,N)
      GO TO 637
C
C     COMPUTE RANDOM STARTING VECTOR

  777 XNORM=0.D0
      DO 3301 I=1,N
      CALL RANDU(IX,IX,R)
      X(I)=.5D0-R
 3301 XNORM=XNORM+X(I)**2
      XNORM=1.D0/DSQRT(XNORM)

C     NORMALIZE VECTOR

      DO 333 I=1,N
  333 X(I)=X(I)*XNORM
  999 DO 692 K=1,II
      SUM=0.D0
      DO 691 J=1,N
  691 SUM=SUM+X(J)*Z(J,K)
  692 TEMP(K)=SUM
      DO 790 I=1,N
      SUM=0.D0
      DO 795 JJ=1,II
  795 SUM=SUM-TEMP(JJ)*Z(I,JJ)
  790 X(I)=X(I)+SUM
      IF(RANDOM) GO TO 637
```

```
      IF(OSC) GO TO 15
      DOUBLE=.TRUE.
      GO TO 15

C     BACK TRANSFORMS VECTORS TO FULL MATRIX

 7165 CALL TRBAK1(L,N,S,E,N,Z)
      DO 654 I=1,N
 654  W(I)=W(I)-KC
      RETURN
      END
```

# XIII. APPENDIX B: IMPLEMENTATION OF THE SYSTEM SOLVER

```
      SUBROUTINE SOLVE(D,E,X,SAVXLM,N,L,TEMP,OSC,C)
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*8 D(L),E(L),X(L),TEMP(L),P2(10),HL(10),P3(10),TOP(10),C(L)
      LOGICAL OSC
      X1=X(1)
      NM1=N-1
      NM2=N-2
      PIVEL=1.D-50
      B=-XLAM*X1
      IF(DABS(B).LT.PIVEL) B=DSIGN(PIVEL,B)
      IF(.NOT.OSC) GO TO 7182
      XLAM=SAVXLM
      OSC=.FALSE.
7182  HL(2)=X(2)*(D(1)-XLAM)-C(2)
      IF(DABS(HL(2)).LT.PIVEL) HL(2)=DSIGN(PIVEL,HL(2))
      P3(2)=(X1*(D(2)-XLAM)-X(2)*E(2))/HL(2)
      TOP(1)=D(1)-XLAM-X1*TEMP(1)
      TOP(2)=E(2)-X1*TEMP(2)+TOP(1)*P3(2)
      DO 7184 I=3,N
      P2(I)=(E(I)*X(I-2))/HL(I-1)
      HL(I)=X(I)*(D(I-1)-XLAM+E(I-1)*P3(I-1))-C(I)
      IF(DABS(HL(I)).LT.PIVEL) HL(I)=DSIGN(PIVEL,HL(I))
      P3(I)=(X(I-1)*(D(I)-XLAM)-X(I)*(E(I)+E(I-1)*P2(I)))/HL(I)
7184  TOP(I)=-X1*TEMP(I)+TOP(I-2)*P2(I)+TOP(I-1)*P3(I)
      IF(DABS(TOP(N)).LT.PIVEL) TOP(N)=DSIGN(PIVEL,TOP(N))
      X(N)=B/TOP(N)
      X(NM1)=P3(N)*X(N)
      DO 7194 J=1,NM2
      I=N-J
7194  X(I-1)=P3(I)*X(I)+P2(I+1)*X(I+1)
      RETURN
      END
```